FIG. 1

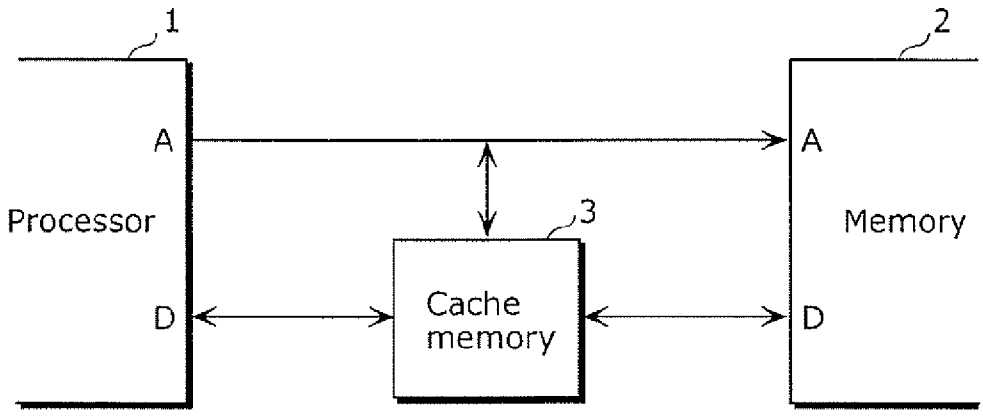


FIG. 2

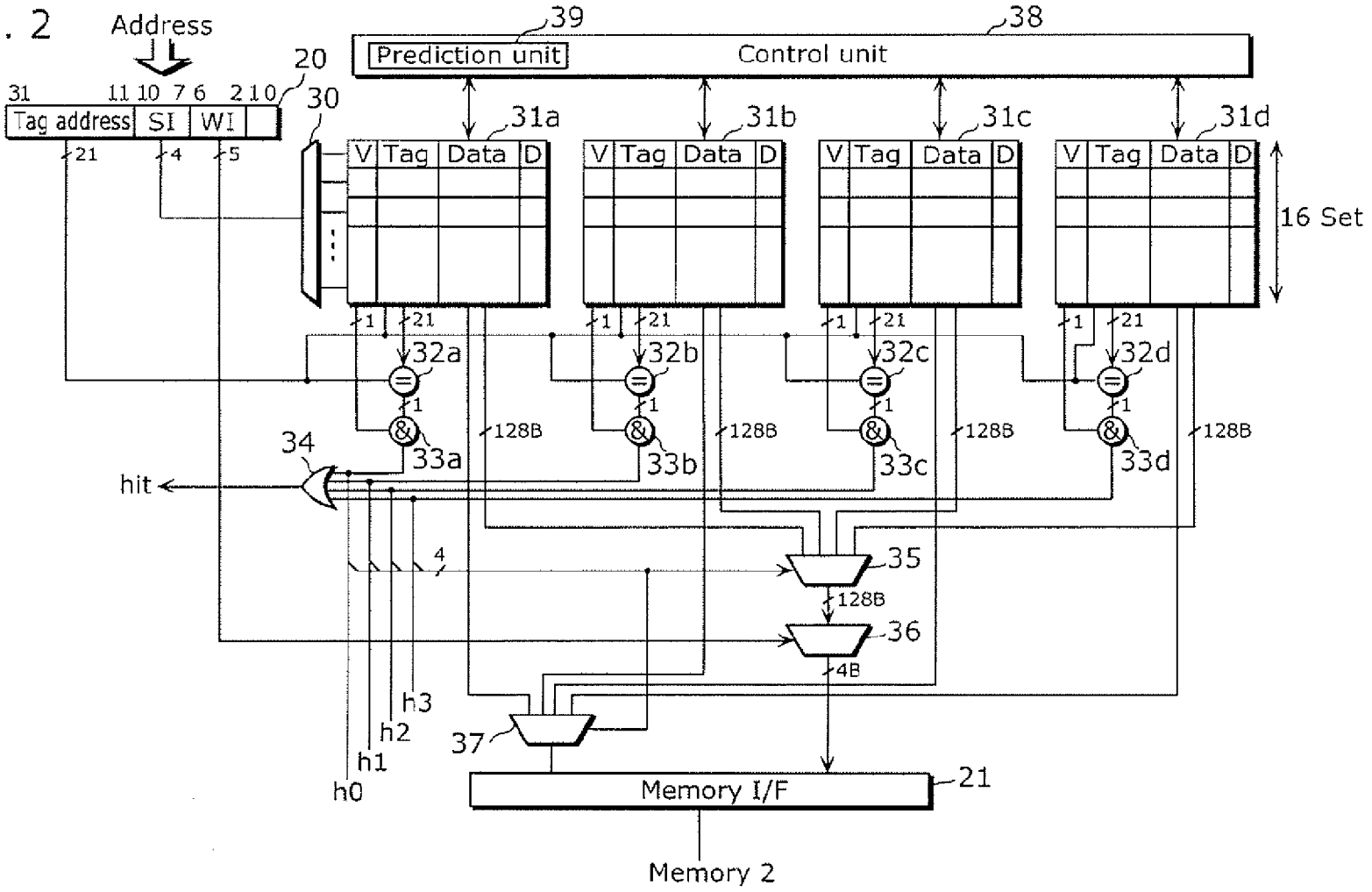

Address
39
38
Prediction unit
Control unit
20
30
31
11 10 7 6 2 1 0
Tag address
SI
WI
21
4
5
31a
31b
31c
31d
V
Tag
Data
D
16 Set
1
21
32a
32b
32c
32d
33a
33b
33c
33d
128B
34
hit
4
35
36
4B
37
h0
h1
h2
h3
Memory I/F
21
Memory 2

FIG. 3

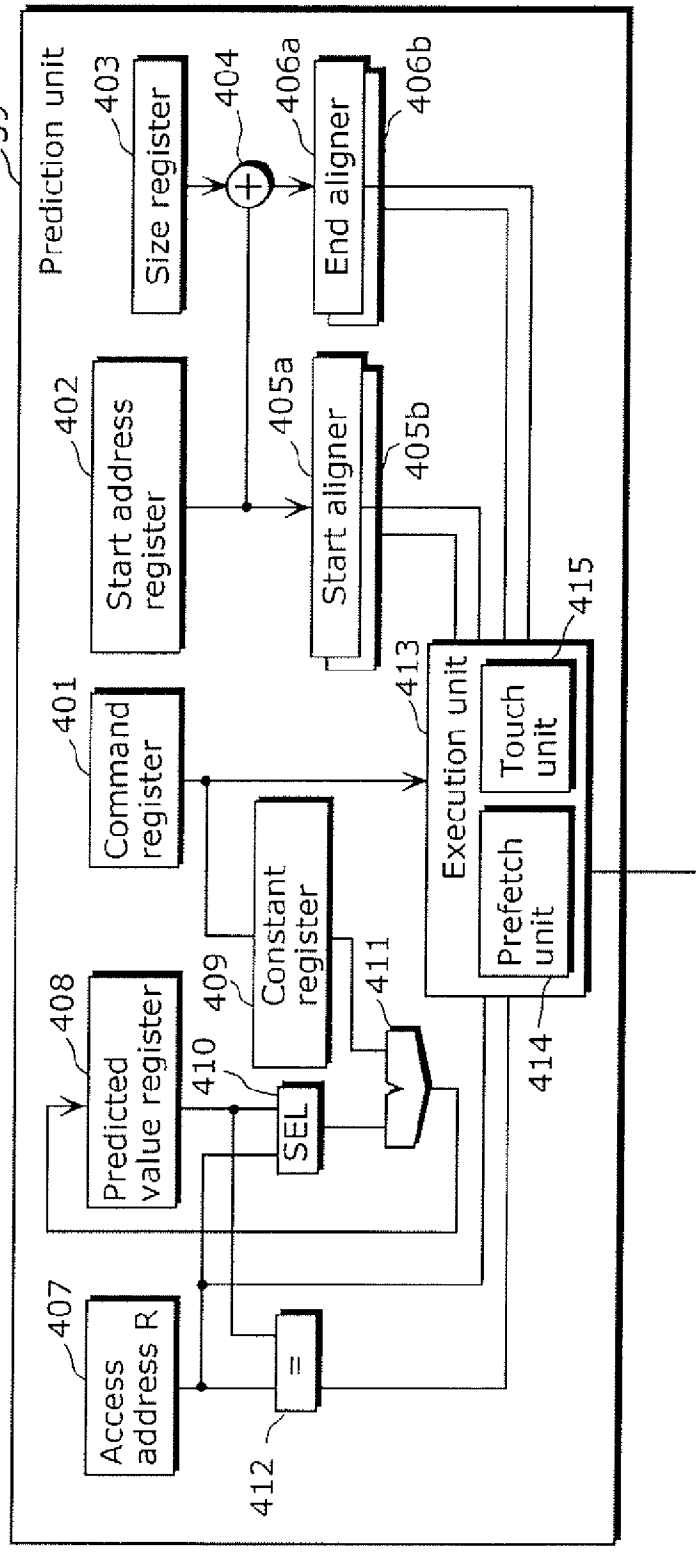

39
Prediction unit
403
Size register
404
406a
End aligner
406b
402
Start address register
405a
Start aligner
405b
401
Command register
413
Execution unit
Touch unit
415
Prefetch unit
414
408
Predicted value register
409
Constant register
411
410
SEL
407
Access address R
=
412

FIG. 4(a)

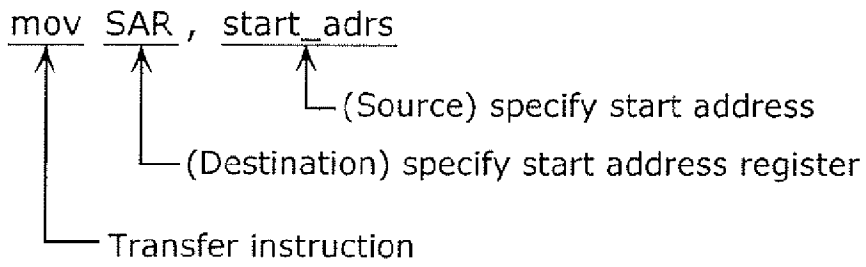


FIG. 4(b)

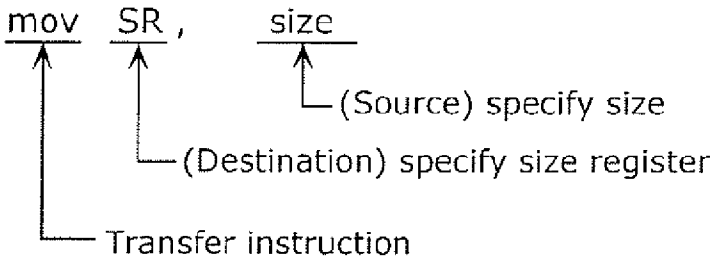


FIG. 4(c)

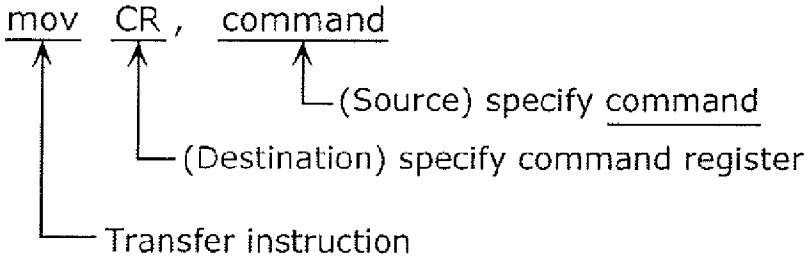


FIG. 4(d)

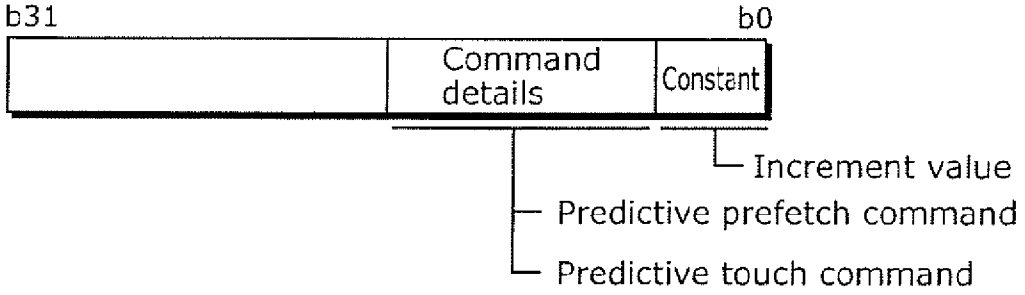


FIG. 5

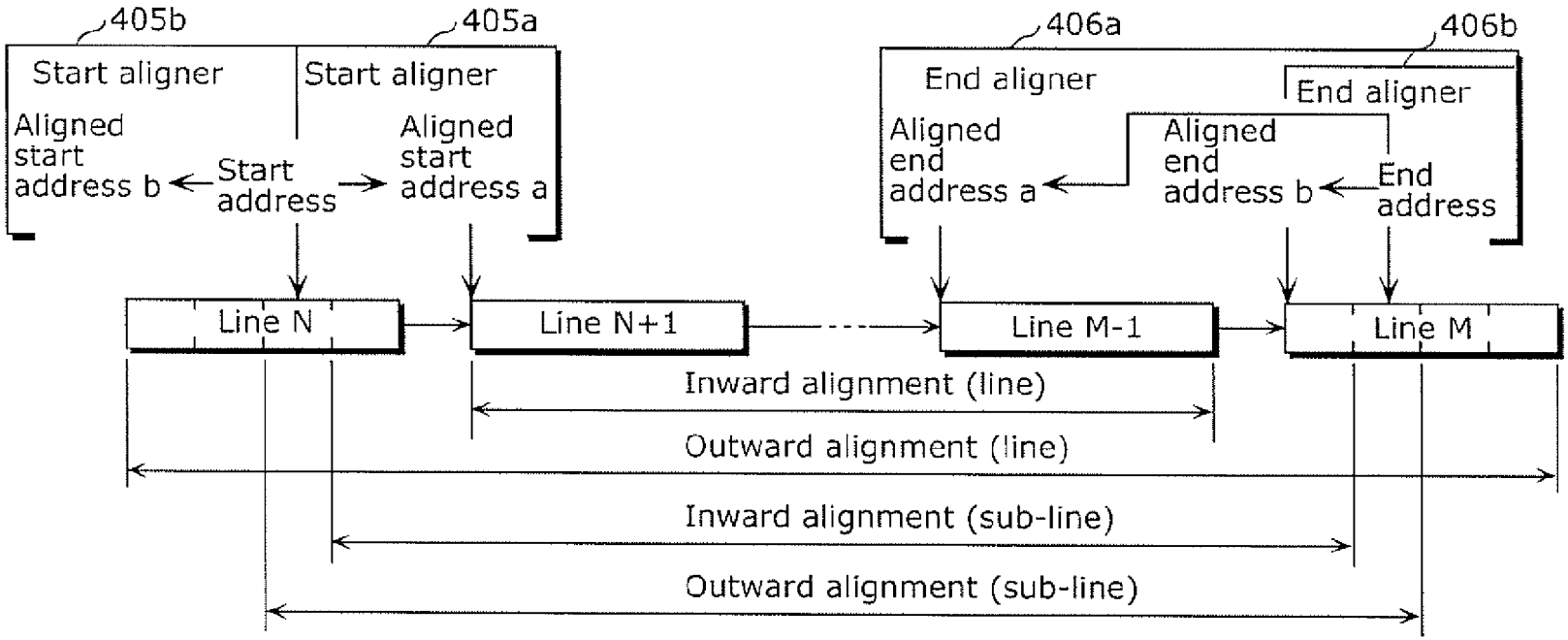

405b
405a
406a
406b
Start aligner
Start aligner
End aligner
End aligner
Aligned start address b
Start address
Aligned start address a
Aligned end address a
Aligned end address b
End address
Line N
Line N+1
Line M-1
Line M
Inward alignment (line)
Outward alignment (line)
Inward alignment (sub-line)
Outward alignment (sub-line)

FIG. 6(a)

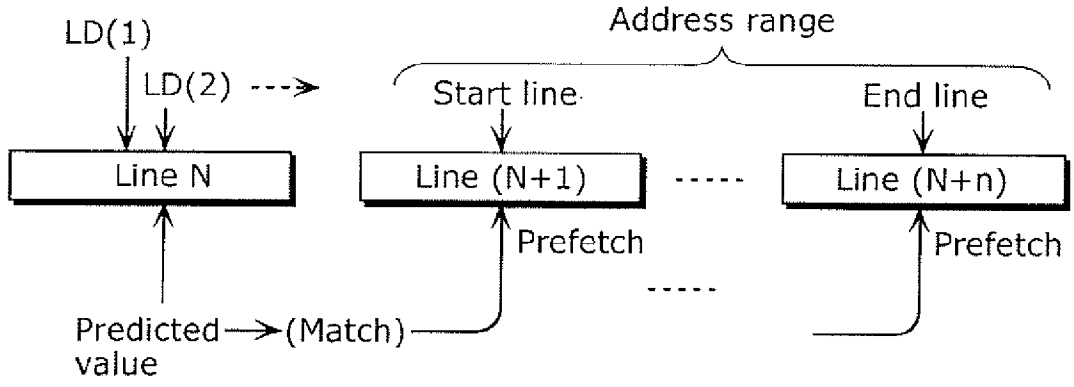

LD(1)
LD(2)
Address range
Start line
End line
Line N
Line (N+1)
Line (N+n)
Prefetch
Prefetch
Predicted value
(Match)

FIG. 6(b)

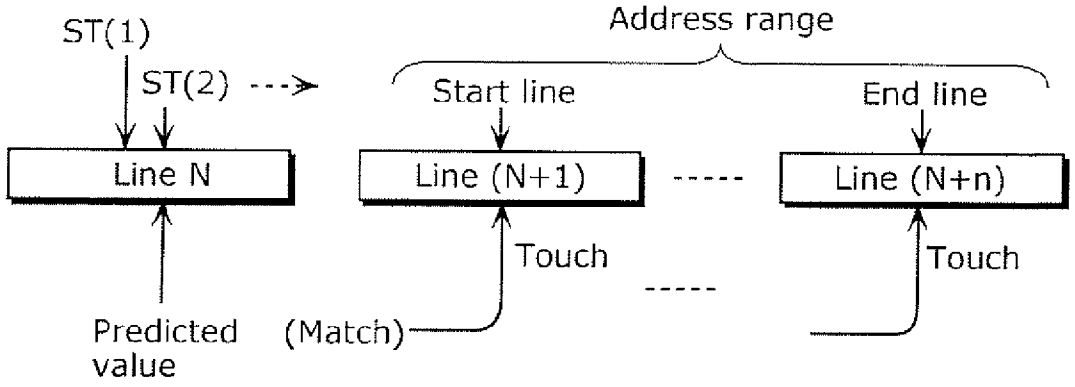

ST(1)
ST(2)
Address range
Start line
End line
Line N
Line (N+1)
Line (N+n)
Touch
Touch
Predicted value
(Match)

FIG. 7

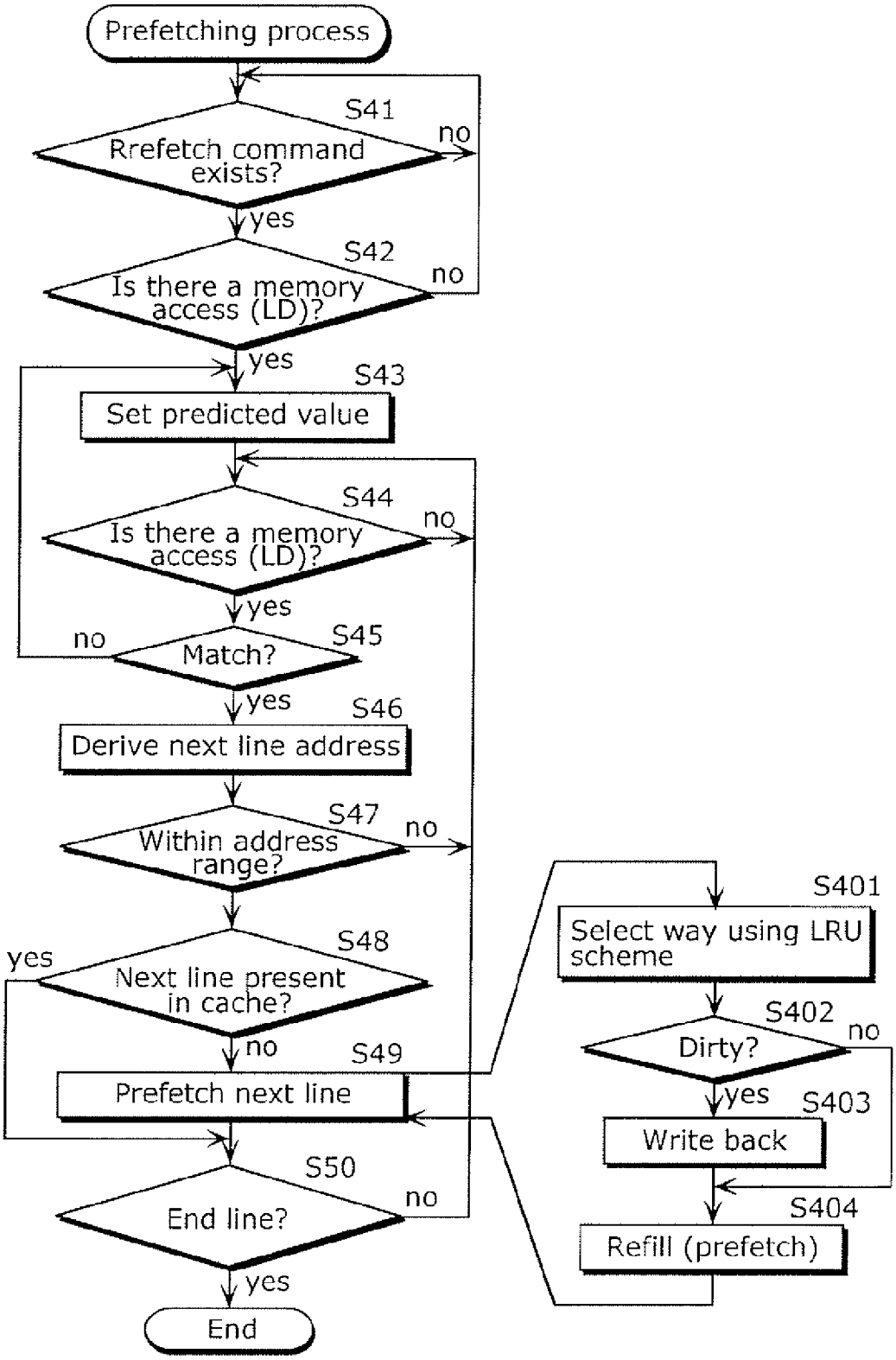

Prefetching process
S41
Rrefetch command exists?
no
yes
S42
Is there a memory access (LD)?
no
yes
S43
Set predicted value
S44
Is there a memory access (LD)?
no
yes
no
S45
Match?
yes
S46
Derive next line address
S47
Within address range?
no
S48
yes
Next line present in cache?
no
S49
Prefetch next line
S50
End line?
no
yes
End
S401
Select way using LRU scheme
S402
Dirty?
no
yes
S403
Write back
S404
Refill (prefetch)

FIG. 8

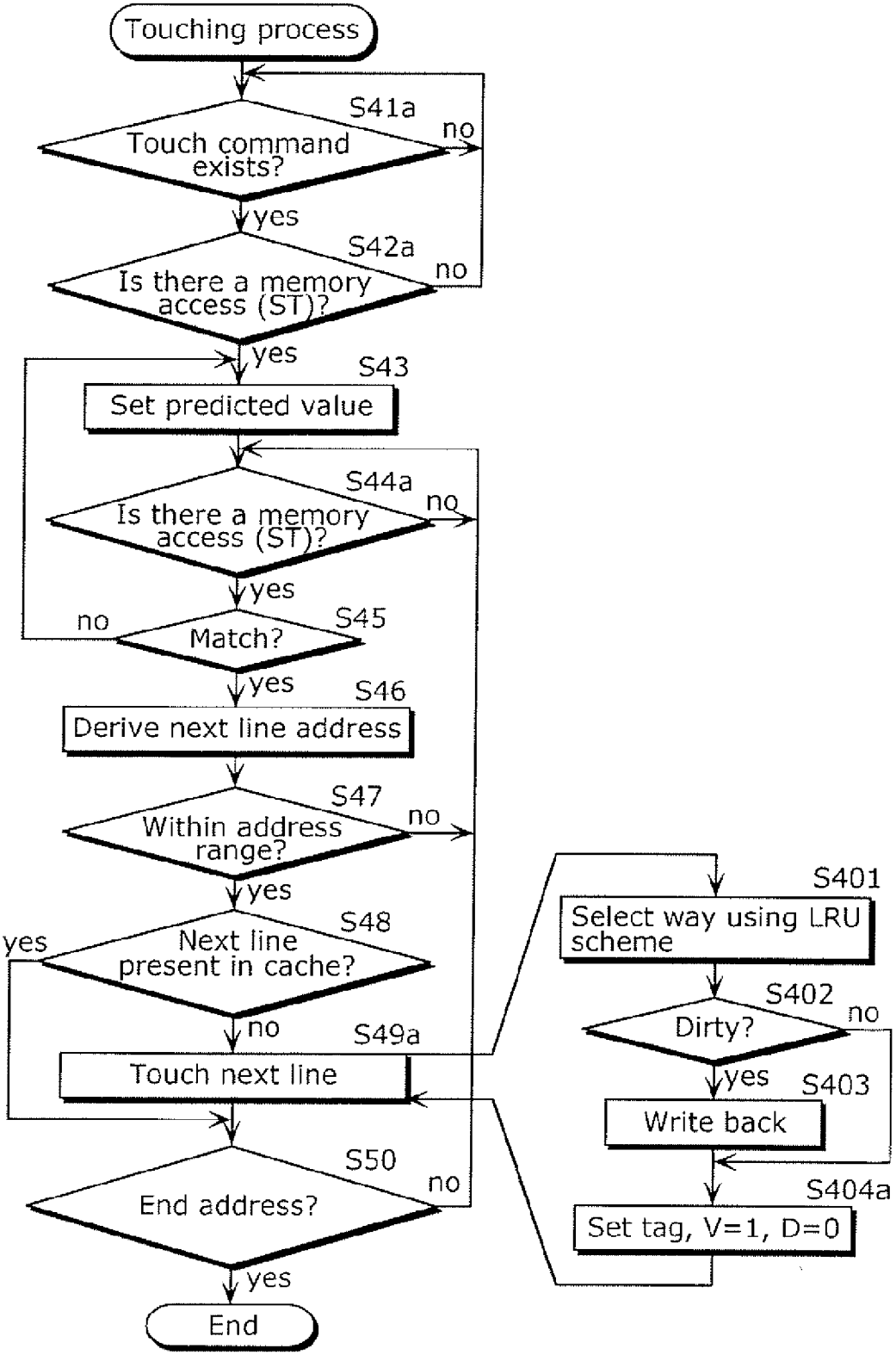

Touching process
S41a
Touch command exists?
no
yes
S42a
Is there a memory access (ST)?
no
yes
S43
Set predicted value
S44a
Is there a memory access (ST)?
no
yes
no
S45
Match?
yes
S46
Derive next line address
S47
Within address range?
no
yes
S48
Next line present in cache?
yes
no
S49a
Touch next line
S50
End address?
no
yes
End
S401
Select way using LRU scheme
S402
Dirty?
no
yes
S403
Write back
S404a
Set tag, V=1, D=0

FIG. 9

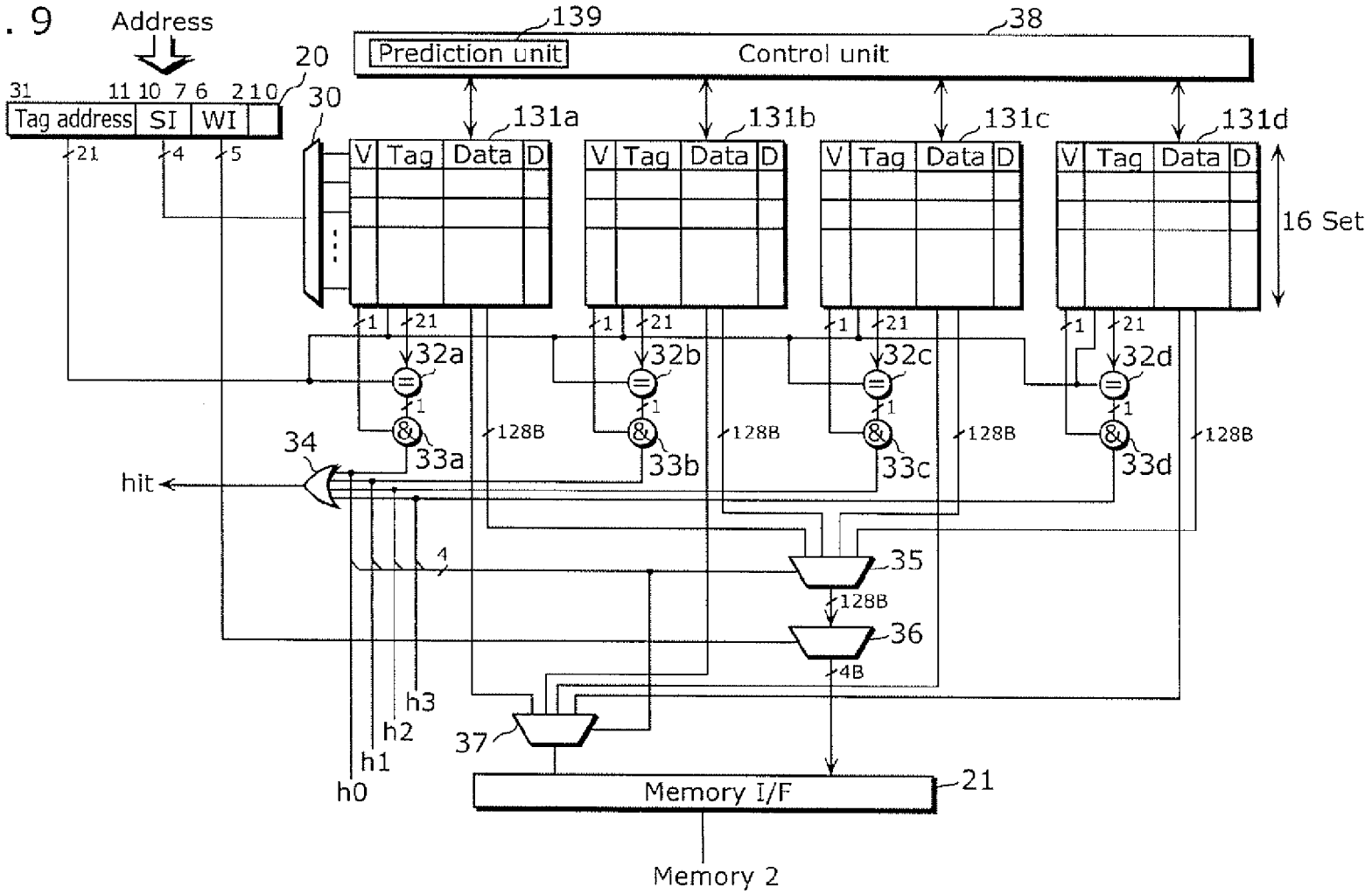

Address
31
11 10 7 6 2 1 0
Tag address
SI
WI
20
30
21
4
5
139
Prediction unit
38
Control unit
131a
131b
131c
131d
V
Tag
Data
D
16 Set
1
21
32a
32b
32c
32d
=
&
33a
33b
33c
33d
128B
34
hit
4
35
36
4B
37
h0
h1
h2
h3
Memory I/F
21
Memory 2

FIG. 10

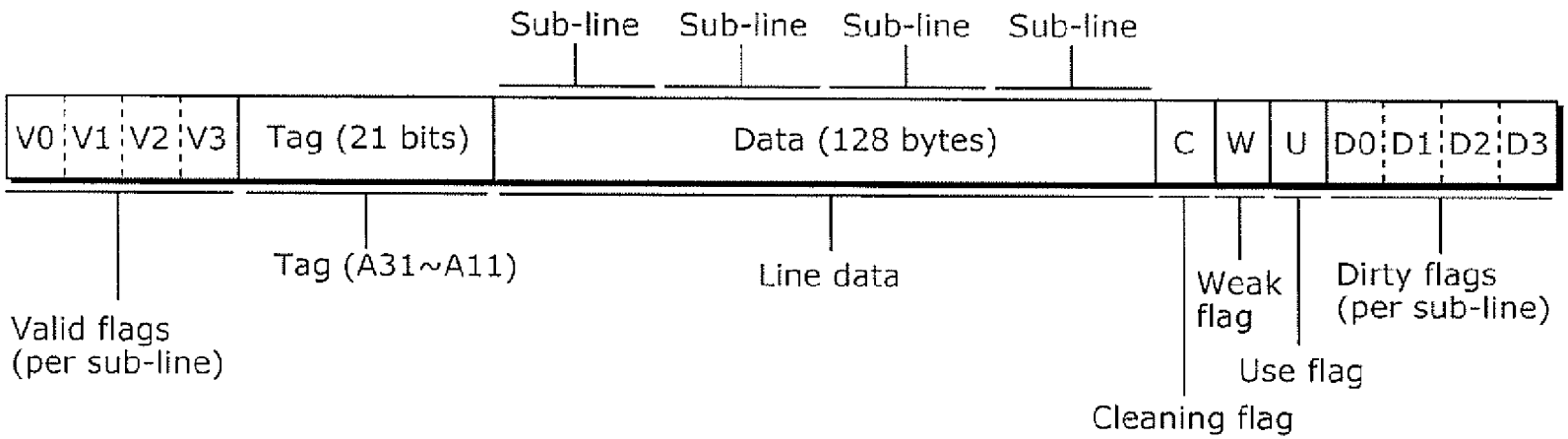

Sub-line
Sub-line
Sub-line
Sub-line
V0
V1
V2
V3
Tag (21 bits)
Data (128 bytes)
C
W
U
D0
D1
D2
D3
Valid flags
(per sub-line)
Tag (A31~A11)
Line data
Weak
flag
Use flag
Cleaning flag
Dirty flags
(per sub-line)

FIG. 11

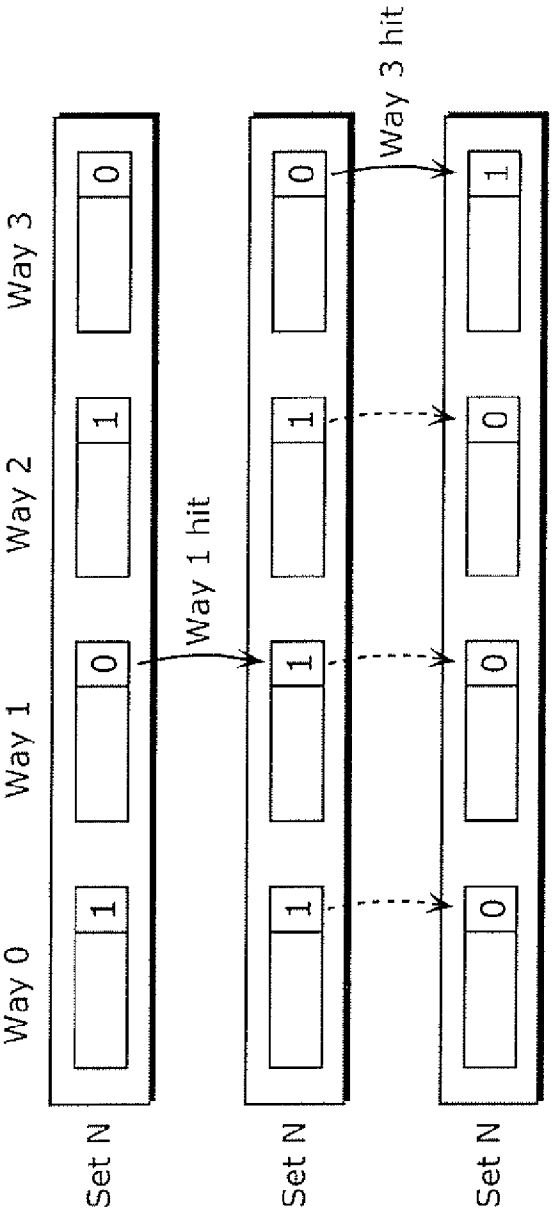

Way 0
Way 1
Way 2
Way 3
Set N
Set N
Set N
Way 1 hit
Way 3 hit
1
0
1
0
1
0
1
0
1
0
1
0

FIG. 12(a)

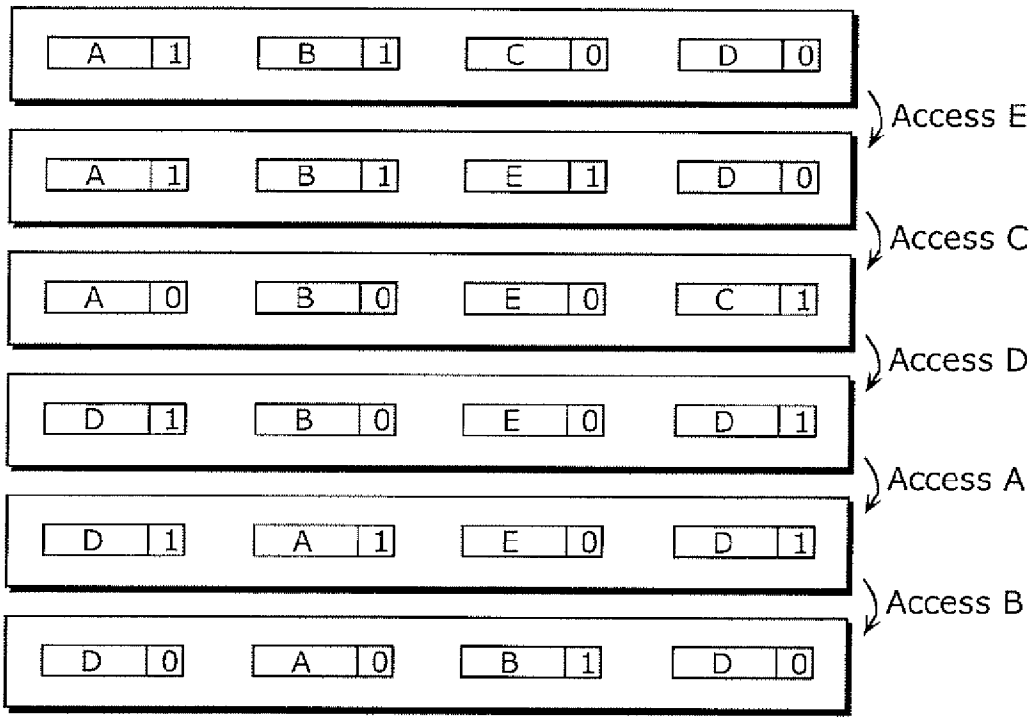


FIG. 12(b)

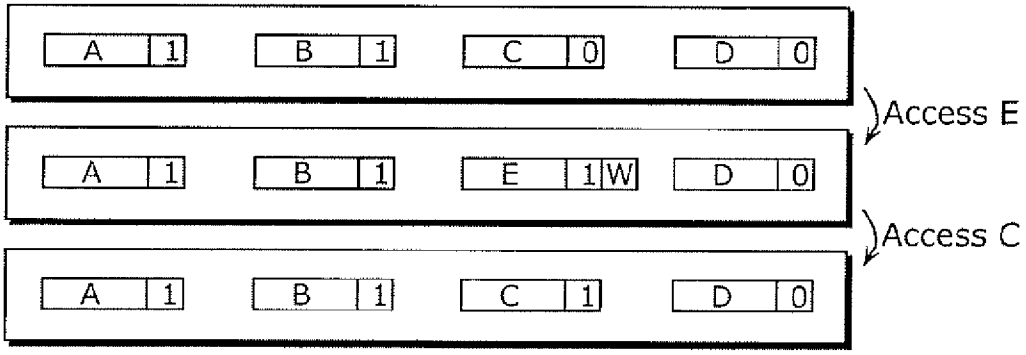


FIG. 13

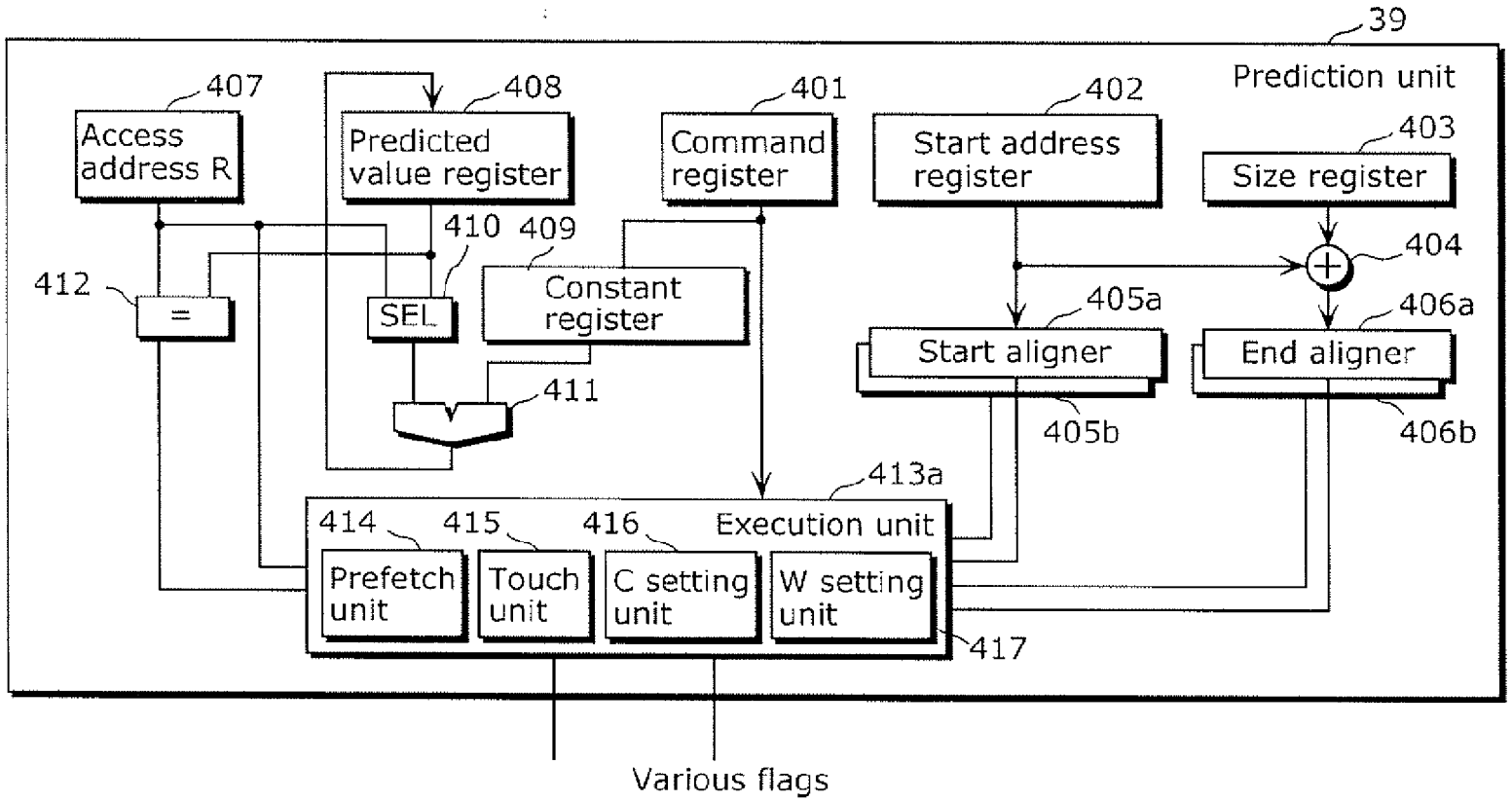

39
Prediction unit
407
Access address R
408
Predicted value register
401
Command register
402
Start address register
403
Size register
404
410
409
412
=
SEL
Constant register
405a
406a
Start aligner
End aligner
411
405b
406b
413a
414
415
416
Execution unit
Prefetch unit
Touch unit
C setting unit
W setting unit
417
Various flags

FIG. 14(a)

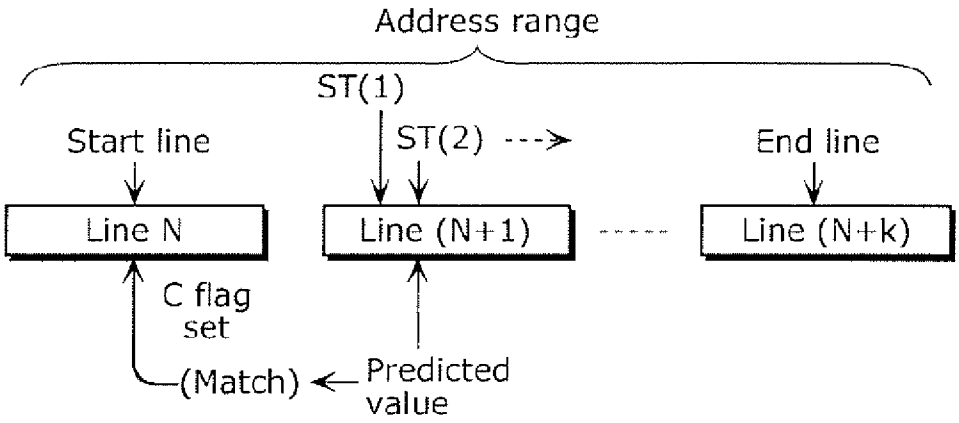


FIG. 14(b)

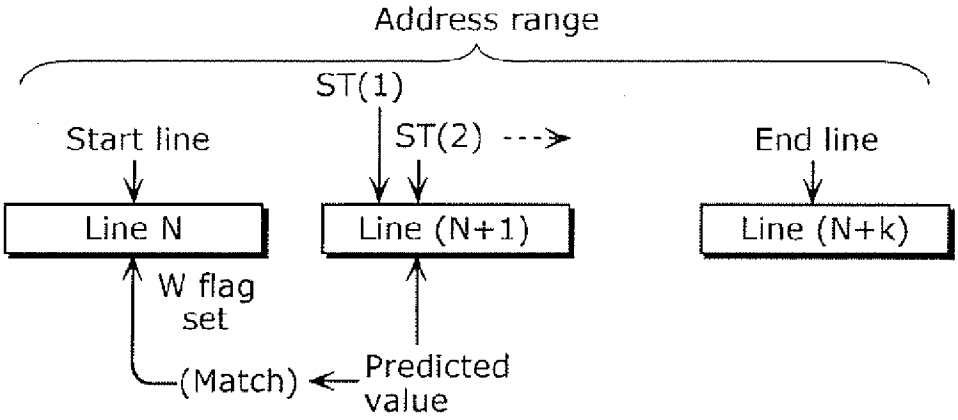


FIG. 15

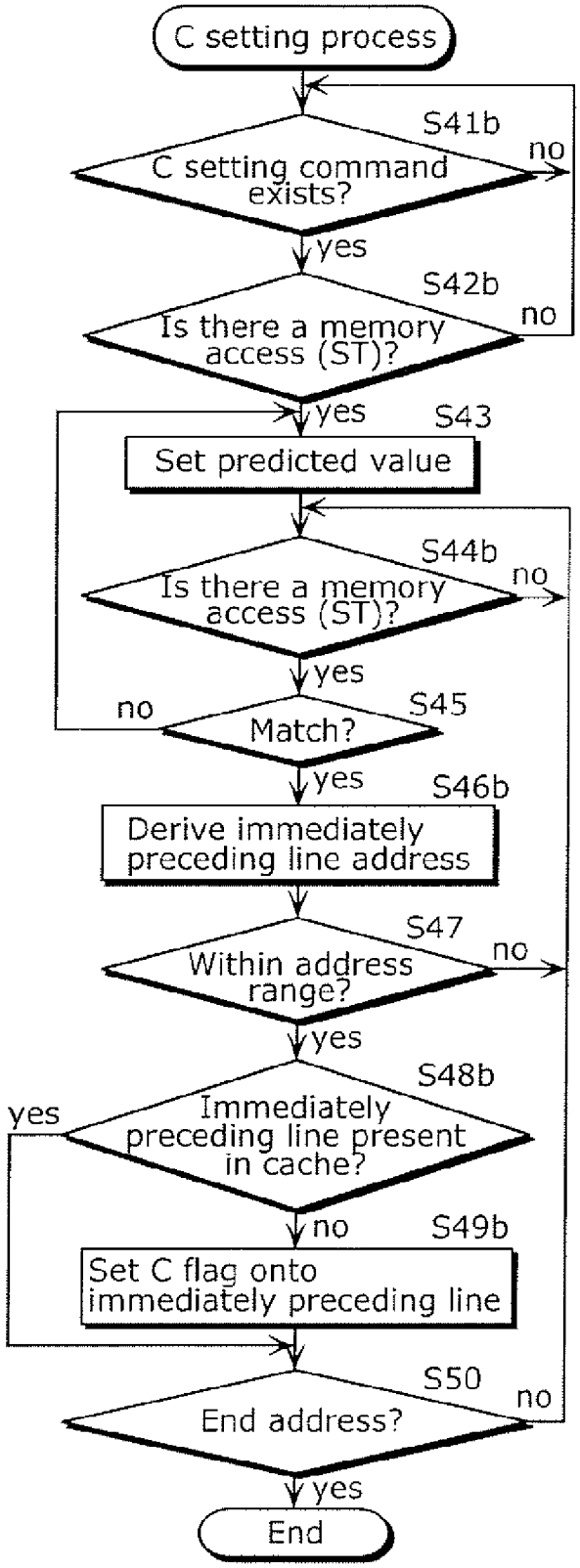

C setting process
S41b
C setting command exists?
no
yes
S42b
Is there a memory access (ST)?
no
yes
S43
Set predicted value
S44b
Is there a memory access (ST)?
no
yes
no
S45
Match?
yes
S46b
Derive immediately preceding line address
S47
Within address range?
no
yes
S48b
yes
Immediately preceding line present in cache?
no
S49b
Set C flag onto immediately preceding line
S50
End address?
no
yes
End

FIG. 16

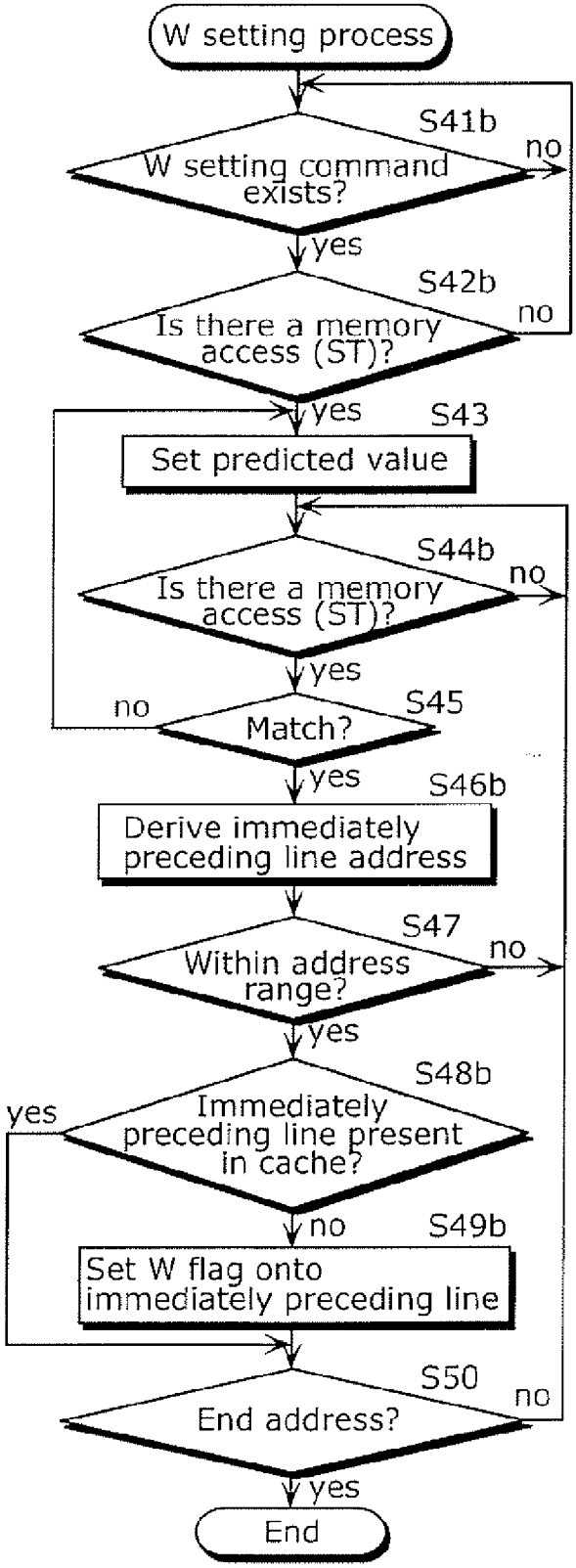

W setting process
S41b
W setting command exists?
no
yes
S42b
Is there a memory access (ST)?
no
yes
S43
Set predicted value
S44b
Is there a memory access (ST)?
no
yes
no
S45
Match?
yes
S46b
Derive immediately preceding line address
S47
Within address range?
no
yes
S48b
yes
Immediately preceding line present in cache?
no
S49b
Set W flag onto immediately preceding line
S50
End address?
no
yes
End

FIG. 17

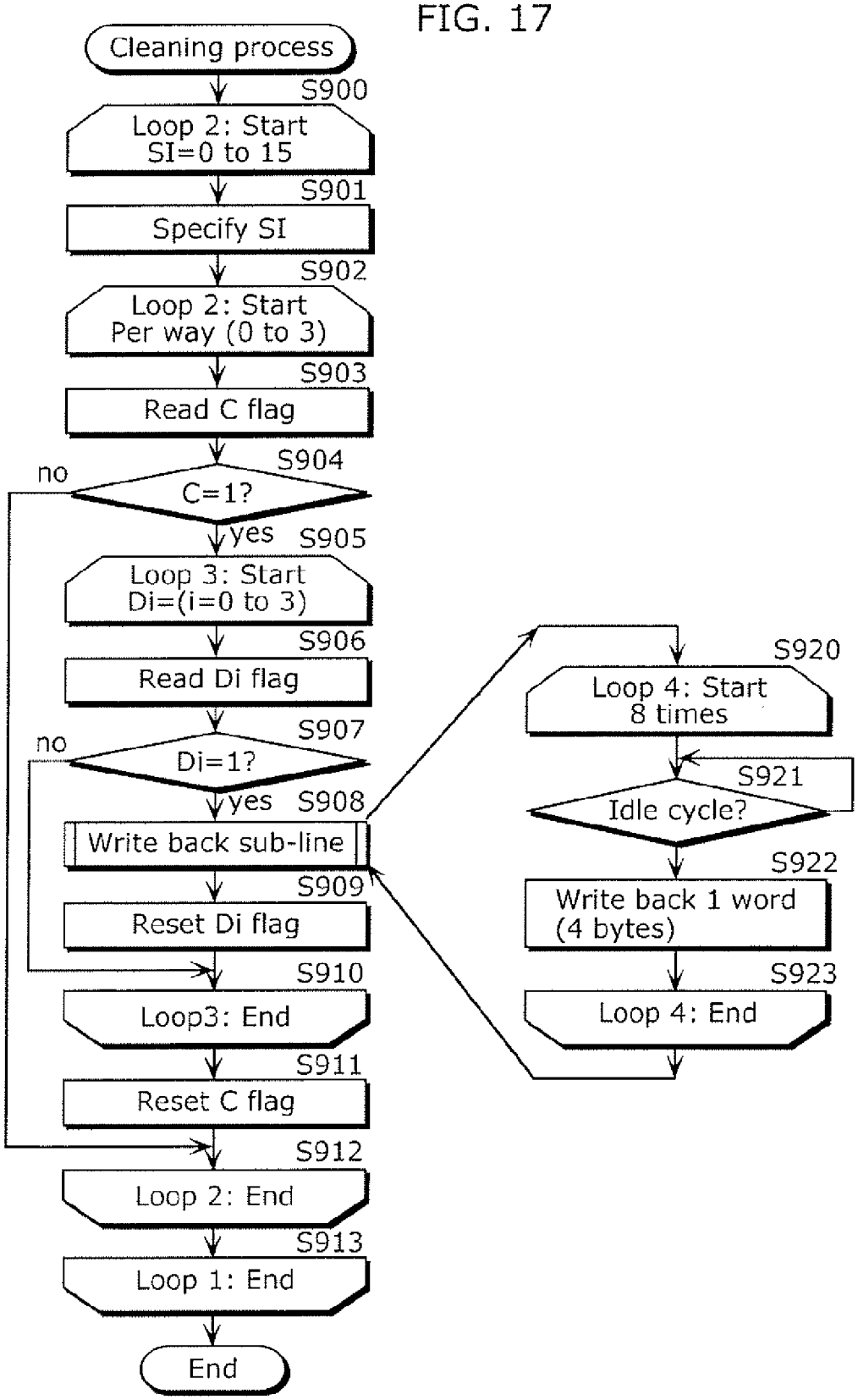

Cleaning process
S900
Loop 2: Start
SI=0 to 15
S901
Specify SI
S902
Loop 2: Start
Per way (0 to 3)
S903
Read C flag
S904
no
C=1?
yes
S905
Loop 3: Start
Di=(i=0 to 3)
S906
Read Di flag
S907
no
Di=1?
yes
S908
Write back sub-line
S909
Reset Di flag
S910
Loop3: End
S911
Reset C flag
S912
Loop 2: End
S913
Loop 1: End
End
S920
Loop 4: Start
8 times
S921
Idle cycle?
S922
Write back 1 word
(4 bytes)
S923
Loop 4: End

FIG. 18

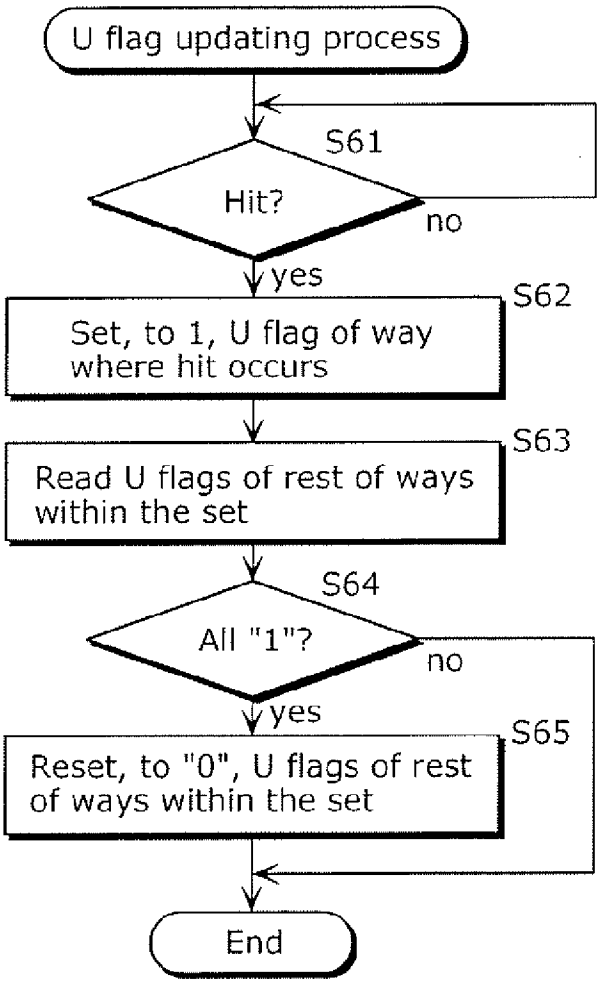
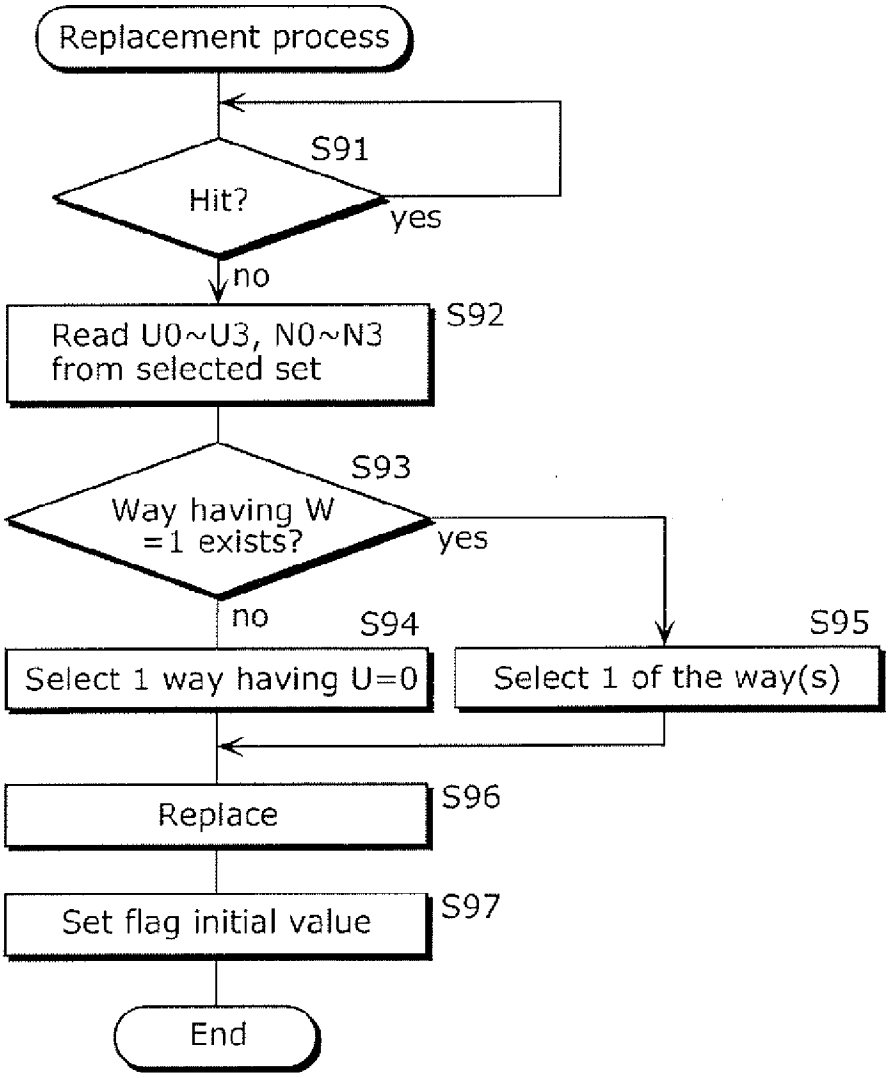

U flag updating process
S61
Hit?
no
yes
S62
Set, to 1, U flag of way where hit occurs
S63
Read U flags of rest of ways within the set
S64
All "1"?
no
yes
S65
Reset, to "0", U flags of rest of ways within the set
End

FIG. 19


Replacement process
S91
Hit?
yes
no
Read U0~U3, N0~N3 from selected set
S92
S93
Way having W =1 exists?
yes
no
S94
S95
Select 1 way having U=0
Select 1 of the way(s)
Replace
S96
Set flag initial value
S97
End